dun & bradstreet

Using Dun & Bradstreet US Equity Alpha Factor Library and Machine Learning to Improve Stock Portfolio Returns

By Chang Lin, Analytics Innovation, Global Data & Analytics

linch@dnb.com

EXECUTIVE SUMMARY

We applied machine learning to model equity future beta adjusted returns using Dun & Bradstreet data attributes . With 400+ factors, we can pick the worst performing stocks with reasonable success, over 16 quarters (2014 – 2017). The results compare favorably with benchmarks using single factors (both public and proprietary).

Overall, we believe Dun & Bradstreet data is a unique data source that, with complete coverage on public stocks in the US, provides extra information not available in other alternative data sources.

INTRODUCTION

Previously, we selected Dun & Bradstreet data attributes[1] based on equity performance separation by single factor decile ranking (Chang Lin, 2018). The result was a collection of factors that, for the past 13 years (Nov. 2004 – Dec. 2017), have provided significant alphas not explained by Carhart's 4-Factor Model (i.e. Fama-French 3-Factor Model plus Momentum). In this study, we go beyond single factor decile ranking, and apply machine learning to select a group of factors that, individually may be weak, but collectively provide strong forecast power on future equity returns. We show how the models perform both in and out sample. The purpose of the study is to provide a guidance on how to incorporate Dun & Bradstreet data attributes into quantitative strategies that need to produce signals on a periodic interval (monthly).

DATA & MODELS

The universe is S&P Total Market Index. Dun & Bradstreet data is available monthly.

Dependent variable Y:

For a given month t, we calculated the next month excess return for all the stocks in the universe and converted it into a categorical variable (1 = top 33.3%, 0 = middle 33.3%, -1 = bottom 33.3%). The look back window is 18 months[2], including current month t (t-17, t-16, …, t). By using a categorical dependent variable, our problem became a classification problem.

Explanatory variables:

There are three D&B datasets included in the test data:

1. CSAD + Standard Scores (Credit Score Archive Database, 191 attributes, 11/2004 – 12/2017)
2. DTRI (Detailed Trade Risk Index, 181 attributes, 12/2010 – 12/2017)
3. Inquiry (Business Inquiry, 62 attributes, 11/2004 – 12/2017)

Each month, we have observed D&B attributes (or factors), for all stocks in the universe. When including both raw and derived attributes, the total number of factors is around 430. Derived attributes are mathematical transformation of raw attributes and can be identified by the suffix in the variable name. For example, "*hicdtavg.pc12*" is derived from "*hicdtavg*" or "high credit average" in CSAD. There are three transformations:

.pc = 1-month percentage

.pc12 = 12-month percentage change

.adj = adjusted (or divide) by sales

[1] Throughout the study, we use "attribute," "factor," or "feature" interchangeably.

[2] The look back window is arbitrarily chosen at 18 months.

Models:

Because there are 400+ factors, some highly correlated, our goal was to pick a subset.

We applied the following machine learning models:

1. Random Forest: the model produces an ensemble of trees that are averaged together to avoid overfitting and reduce variance. Also, each tree only works on a subset of the observations and features, minimizing the influence of some highly influential features. Overall, it should produce a robust result. We used the Python implementation from *scikit-learn* module.

2. Gradient Boosting: the model also produces an ensemble of weak prediction models, typically decision trees. In contrast to Random Forest, trees are produced sequentially and added to existing ensemble, with each new tree trying to (slowly) correct the errors of current ensemble model. The implementation we use is *LightGBM*, a high-performance gradient boosting algorithm in Python.

3. Logistic Regression: this is a linear model. Because many factors are correlated, L1 regularization (Lasso) is applied to handle collinearity. Another benefit of L1 regularization is automatic feature selection. The implementation we use is the *glmnet* library in R.

Data preprocessing and cleaning, missing data imputation:

If a factor has too much missing data cross sectionally (70% missing), we removed the factor from this study. Otherwise, for each month, we imputed the missing factor to zero, then transformed each factor to its cross-sectional ranking (from 0 to 1.0). Another popular transformation is normalization by each factor's cross-sectional mean and standard deviation. Although ranking loses the original structure, it handles outliers better in practice. This may not matter for non-linear models such as Random Forest, but it does matter for the generalized linear models that we are going to apply here.

Model Approach

During in-sample model fitting, we removed middling samples, where dependent variable Y = 0. The motivation is that middle samples tend to confuse models, and models can be better trained with more extreme samples. As a result, our problem becomes a binary classification. When it comes to prediction, instead of predicting binary outcome, we used predicted probability to rank stock performance, and again divide stocks into 3 classes (1, 0, -1) based on predicted probability.

CROSS VALIDATION AND FEATURE IMPORTANCE

We chose January 2014 (201401) for model fitting and tuning. All three models have some parameters to be tuned for optimal performance. Here we refrained from this practice as our goal was to show the values of Dun & Bradstreet factors. The exception is the penalized logistic regression (in *glmnet*), where automatic tuning in λ is performed. If model performance can be further improved by fine-tuning certain model parameters, all the better, but that is not the goal of this study.

Figure 1 shows the top 20 features selected by *LightGBM*, indicated by the number of times a feature is used in a model (out of 100 trees):

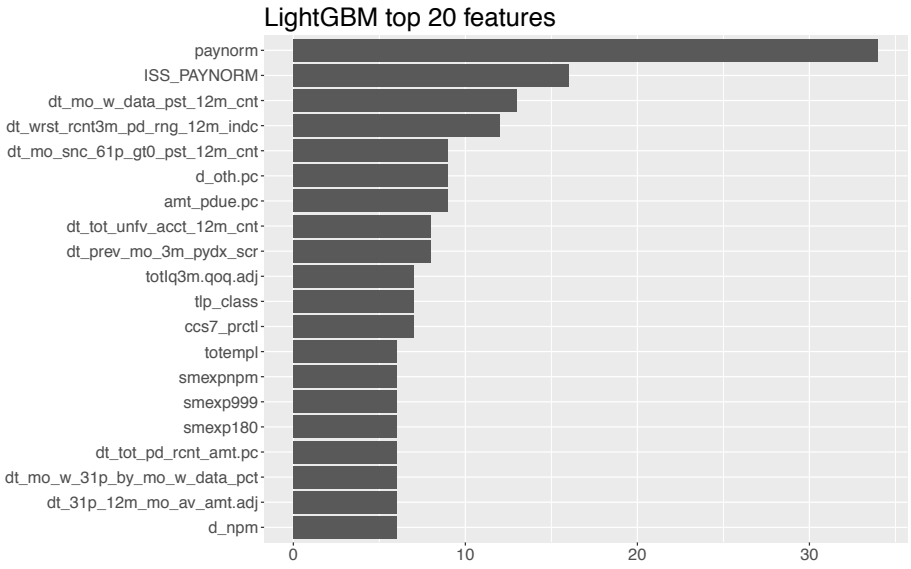

Figure 1 Top 20 features selected by lightGBM during in-sample model fitting

Logistic Regression with L1 (Lasso) penalty.

Because logistic regression is essentially a linear regression on explanatory variables, the only parameter that can be tuned is λ, the penalty coefficient. The larger the λ, the larger the penalty. The optimal λ is again obtained via cross-validation path, around -*log*(λ)=7.5, according to the left chart in Figure 2 below. A nice outcome of Lasso is automatic feature selection. The right chart below shows the top 20 features selected by Lasso[3], which is chosen based on the absolute value of the coefficients on input factors (cross-sectionally ranked).

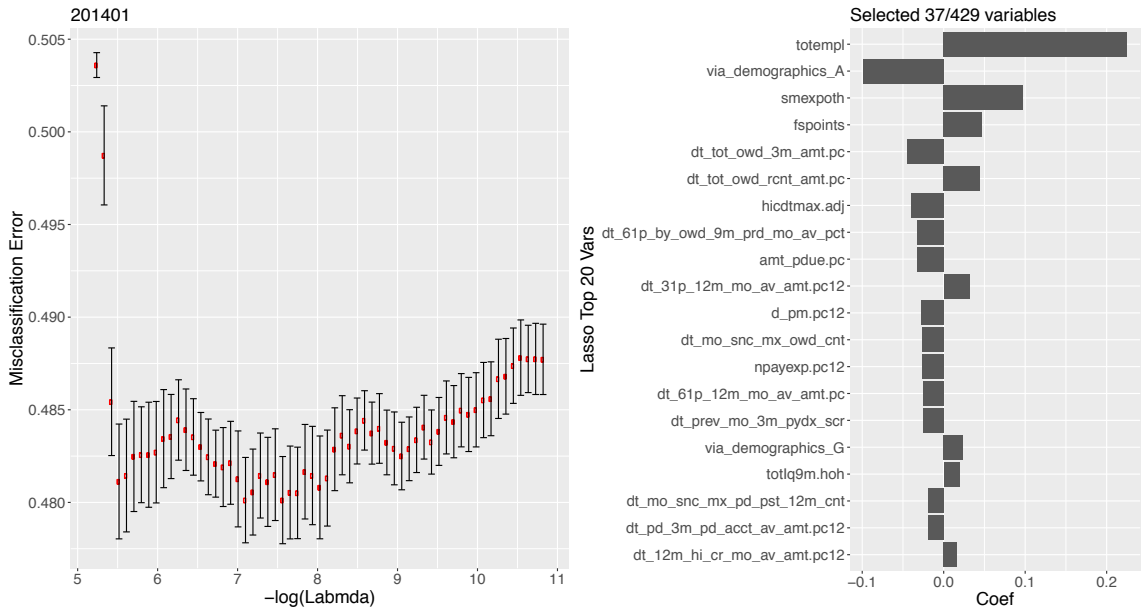

Figure 2 Left: cross-validation path for Logistic Regression with L1 penalty. Optimal λ is chosen so that the minimum misclassification error is achieved. Right: Top 20 features selected with optimal λ.

Random Forest:

There are a few parameters that can be tuned, including the maximum number of features in each tree, maximum depth (interaction), etc. For this algorithm we only kept default values for most parameters, except we set max_depth=3, due to the time it takes to finish each run, with the understanding that the result may not be optimal.

OUT OF TIME PERFORMANCE

For the Out of Time test, we sampled 4 years (2014 – 2017), and 4 quarters within each year (March, June, September, December), for 16 total data points. For each data point, say t_0 = 201403 (March 2014), there is an 18-month look back window, or 18 observations in (Y_t^i, X_t^i) for t = t_0-18,…,t_0-1, and i=1, …, N. We used a fitted model based on the entire 18 observations and produced forecast $\widehat{Y_{t_0}}$. For logistic regression with L1 penalty (Lasso), we used the cross-validated path to choose optimal λ before generating the forecast.

Figure 3 shows ROC (Receiver Operating Curve[4]) for the 3 models, and 3 single factors, based on how well they capture the worst-performing stocks at various probability thresholds. The 3 factors are *net worth* and *sales* (both of which are publicly available) and *ISS_PAYNORM*, which is one of the strongest Dun & Bradstreet factors (Chang Lin, 2018; Paul K. Lieberman, 2017).

Both *net worth* and *sales* perform better than coin flipping or random guesses, and all 3 machine learning models perform better than the two public factors. *ISS_PAYNORM* does not perform well in this study, but we understand that it is not designed to capture bad stocks, but rather to capture a company's leverage power on suppliers (Paul K. Lieberman, 2017). Among 3 models, Random Forest performs the best. Table 1 shows the two-sample k-s test on whether each curve is distinguished from random guess. All of them are significantly different from random guess except for *ISS_PAYNORM*.

[3] As explained earlier, some of attributes have low coverage and are not included in model inputs.

[4] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

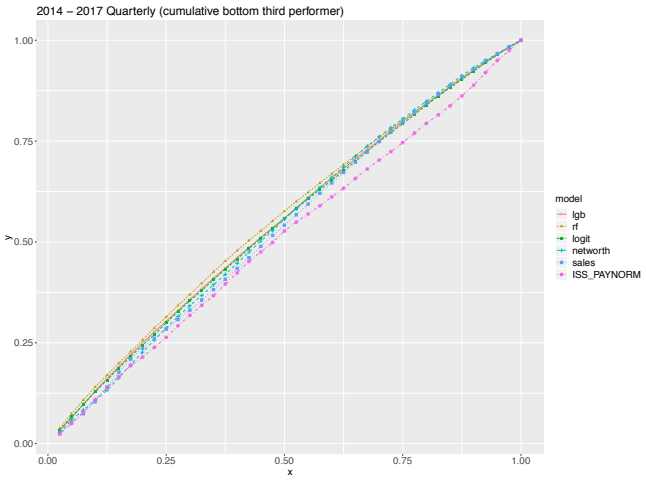

Figure 3 Receiver Operating Curves for how each model and factor captures the percentage of worst performing stocks at various probability threshold. lgb = lightGBM, rf = RandomForest, logit = Logistic Regression with L1 penalty.

Model	k-s statistic	p-value
lgb	0.061	8.32E-07
rf	0.077	1.08E-10
logit	0.057	4.34E-06
networth	0.061	7.37E-07
sales	0.048	2.33E-04
ISS_PAYNORM	0.027	1.19E-01

Table 1 k-s statistic and p-values. When p-value ≤ 0.05, it indicates the associated model or factor produces result significantly different from random guess.

TOP 20 FEATURES (OUT OF TIME, AVERAGED QUARTERLY)

Table 2 shows the average top 20 features by each model out of time, ranking in descending order of importance (1 = most important). We observed that 5 factors appear in all three models (fields below in bold black), although in different order of importance. In general, the factors selected at the top are quite different. This is because many of Dun & Bradstreet's factors are correlated, so models favor one over others, resulting in different subsets selected.

Recall that we have selected 13 factors with strong performance in decile ranking (Chang Lin, 2018), but only a few of them (*hicdtavg.adj*, *ISS_PAYNORM*) appear in the top 20 in multi-factor models. This can be due to different objectives of the two studies: the previous study identified factors that have significant return separation in top and bottom decile ranking; this study identifies the worst performing stocks.

	RandomForest	Gradient Boosting (lightGBM)	Lasso
1	**via_demographics_G**	ISS_PAYNORM	**via_demographics_G**
2	npayexp.adj	**paynorm**	via_demographics_C
3	payref.adj	**totempl**	via_demographics_M
4	smexpsln.adj	via_overall_rating	**totempl**
5	pexp_s_n.adj	**ser**	**fspoints**
6	npayexpp.adj	**fspoints**	via_demographics_N
7	smexpcur.adj	d_neg.pc	via_data_depth_A
8	**paynorm**	**via_demographics_G**	via_demographics_T
9	smexpsat.adj	dt_mo_w_data_pst_12m_cnt	**ser**
10	**ser**	dt_rcnt3m_mx_31p_pst_12m_indc	totlq24m
11	**fspoints**	dt_mo_snc_tot_pd_gt0_pst_12m_cnt	smexpoth
12	via_portfolio_rating	via_portfolio_rating	payref.sta
13	d_slow.adj	dt_rcnt3m_mx_61p_pst_12m_indc	**paynorm**
14	**totempl**	dt_wrst_rcnt3m_pd_rng_12m_indc	dt_3m_hi_cr_mo_mx_amt.adj
15	d_slng.adj	payref.adj	via_data_depth_H
16	hicdtavg.adj	ccs7_prctl	ccs7_prctl
17	totlq24m.qoq.adj	suits.pc12	d_npm.adj
18	dt_pd_3m_acct_av_amt.adj	suits.pc	d_oth.pc
19	ISS_PAYNORM	d_60.pc	via_demographics_E
20	d_pm.adj	dt_rcnt3m_mx_pd_pst_12m_indc	totlq6m.hoh

Table 2 Average Top 20 features selected out of time by each model

Table 3 summarizes Figure 3 at x = 1/3, the predicted probability threshold for labeling the worst-performing stocks (or bads). At this threshold, Gradient Boosting (lgb) captures 39.3% of bads, for a gain of 17.8%. Random Forest (rf) captures 40.8% of bads, for a gain of 22.4%. Logistic Regression with L1 penalty (logit) captures 38.8% of bads, for a gain of 16.3. They all capture more bads than *net worth*, *sales*, and *ISS_PAYNORM*. We used the same probability threshold to split stocks in two groups: bottom group (bottom one-third) and top group (top two-thirds). In terms of returns, the bottom group significantly underperforms the top group, for both equal weighted (rtnBot vs rtnTop) and cap-weighted (rtnwBot vs rtnwTop) beta adjusted returns. (We note that overall beta adjusted returns are negative during the out-of-time period).

				Equal-Weighted			Market-Cap Weighted				
Model	Predicted bads	Captured bads	gain	rtnBot	rtnTop	rtnAll	rtnwBot	rtnwTop	rtnwAll	MCapBot	MCapTop
Lgb	0.333	0.393	0.178	-196.9	-109.9	-138.7	-18.7	-0.9	-3.0	358.4	1245.2
Rf	0.333	0.408	0.224	-224.4	-96.8	-138.7	-71.8	1.9	-3.0	217.0	1528.7
Logit	0.333	0.388	0.163	-184.0	-116.4	-138.7	-12.8	2.8	-3.0	378.3	1171.8
Networth	0.333	0.376	0.127	-185.7	-116.5	-138.7	-42.1	0.8	-3.0	102.9	1568.1
Sales	0.333	0.364	0.093	-185.0	-116.9	-138.7	-61.4	1.7	-3.0	130.2	1636.3
ISS_PAYNORM	0.321	0.339	0.057	-182.4	-118.6	-138.7	-31.8	8.7	-3.0	640.2	893.1

Table 3 all return numbers are in bps (basis points); MCapBot and MCapTop are medium market capitalization for bottom and top groups, respectively, in million dollars.

Figure 4 shows the cumulative spread difference between top and bottom groups, for both equal-weighted and cap-weighted. Random Forest performs the best in both categories. The model captured more bad stocks (which is the goal of the study), which translates to better performance separations.

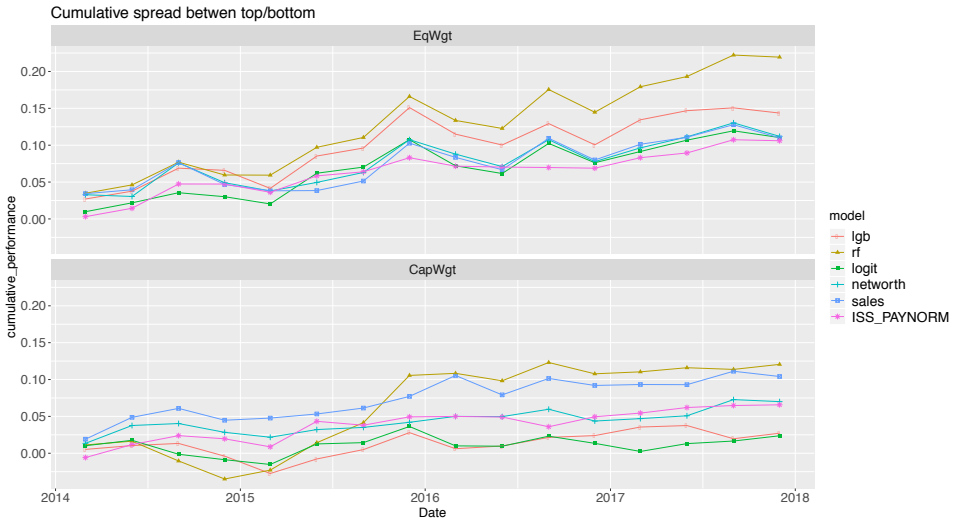

Figure 4 vertical (cumulative_performance) is real return

SUMMARY

Coincidentally, *sales* is related to account receivables on financial statement, *net worth* is the book value of the company or shareholder's equity, while Dun & Bradstreet factors are related to account payables. Together they complement each other to present a complete picture of a company's financial health. Because of this, it makes sense to assume that Dun& Bradstreet factors provide extra signals not captured by *net worth* and *sales,* and the study shows that Dun & Bradstreet's proprietary factors do a better job capturing the worst-performing stocks. Furthermore, Dun & Bradstreet factors are updated on monthly basis, providing more timely and frequent signals than quarterly financial statements. Through more creative feature engineering, and more sophisticated machine learning models, we believe Dun & Bradstreet factors can be even more valuable to quantitative funds and asset managers.

REFERENCE:

1. Chang Lin, Dun & Bradstreet's US Equity Alpha Factor Library, May 2018 (available upon request)
2. Paul K. Lieberman, Capital Markets Study: Payment Power and Cross Section of Stock Returns, May 2017